

Claims

WHAT IS CLAIMED IS:

1. A computer program product encoding a computer program for executing on a computer system a computer process for dynamically generating typing context data associated with a typing-context-relevant-code-point being executed within a typing context in a dynamic execution environment, the computer process comprising:

5 encountering the typing-context-relevant-code-point in the typing context during execution of the program;

 identifying a typing context handle associated with the typing context, the typing context handle referencing a typing context data structure associated with the typing context;

 computing the typing context data associated with the typing-context-relevant-code-point;

10 allocating a field in the typing context data structure associated with the typing-context-relevant-code-point; and

 recording the typing context data in the field of the typing context data structure.

2. The computer program product of claim 1 wherein the typing-context-relevant-code-point executes a type test on an instance of a generic class, the typing context data includes a resource type descriptor defining the exact type of the instance, and the computer process further comprises:

5 performing the type test based on the resource type descriptor associated with the typing-context-relevant-code-point.

3. The computer program product of claim 1 wherein the typing-context-relevant-code-point executes an allocation of an instance of a generic class, the typing context data includes a resource type descriptor defining the exact type of the instance, and the computer process further comprises:

5 creating the instance of the generic class based on the resource type descriptor associated with the typing-context-relevant-code-point, wherein the instance is of the exact type.

4. The computer program product of claim 1 wherein the typing-context-relevant-code-point calls a generic method, the typing context data includes another typing context handle, and the computer process further comprises:

5 passing the other typing context handle referencing the typing context data to the generic method as a hidden parameter.

5. The computer program product of claim 1 wherein the identifying operation comprises:

retrieving the typing context handle from a stack frame.

6. The computer program product of claim 1 wherein the typing-context-relevant-code-point is executed within an instance of a generic class and the identifying operation comprises:

retrieving a first pointer to the instance; and

5 retrieving the typing context handle via a second pointer, a second pointer being relative to the first point and referencing the typing context handle associated with the instance.

7. The computer program product of claim 1 wherein the computing operation comprises:
- retrieving the typing context data associated with the typing-context-relevant-code-point from a global hash table.
8. The computer program product of claim 1 wherein the encountering operation comprises:
- assigning an index to the typing-context-relevant-code-point.
9. The computer program product of claim 8 wherein the allocating operation comprises:
- allocating the field in the typing context data structure, in accordance with the index.
10. The computer program product of claim 8 wherein the index is assigned based on the “arity” of the typing-context-relevant-code-point.
11. The computer program product of claim 8 wherein the index is assigned based on a category associated with the typing-context-relevant-code-point.
12. The computer program product of claim 11 wherein the category is assigned on a per-containing class basis.
13. The computer program product of claim 11 wherein the category is assigned on a per-containing method basis.
14. The computer program product of claim 11 wherein the category is assigned on a per-containing assembly basis.

15. A computer program product encoding a computer program for executing on a computer system a computer process for accessing typing context data associated with a typing-context-relevant-code-point being executed within a typing context in a dynamic execution environment, the computer process comprising:

- 5 encountering the typing-context-relevant-code-point in typing context during execution;
- identifying a typing context handle associated with the typing context, the typing context handle referencing a typing context data structure associated with the typing context; and
- retrieving from the typing context data structure the typing context data associated with the typing-context-relevant-code-point.

16. The computer program product of claim 15 wherein the computer process further comprises:

 allocating a field in the typing context data structure during execution, responsive to the identifying operation, the field being associated with the typing-context-relevant-code-point.

17. The computer program product of claim 15 wherein the typing context data structure includes a table and the obtaining operation comprises:

- identifying an index associated with the typing-context-relevant-code-point; and
- obtaining typing context data associated with the typing-context-relevant-code-point from
- 5 an indexed slot in the table, based on the index.

18. The computer program product of claim 17 wherein the index is associated with the typing-context-relevant-code-point during execution of the program.

19. The computer program product of claim 15 wherein the encountering operation comprises:

encountering an open-type expression including a type test.

20. The computer program product of claim 15 wherein the encountering operation comprises:

encountering an open-type expression including an allocation of an instance of a generic class.

21. The computer program product of claim 15 wherein the encountering operation comprises:

encountering the open-type expression within a scope of an instance of a generic class.

22. The computer program product of claim 15 wherein the typing context is defined within the scope of a stack frame and the operation of identifying a typing context handle comprises:

reading the typing context handle from the stack frame.

23. The computer program product of claim 15 wherein the typing context is defined within the scope of an instance of a generic class and the operation of identifying a typing context handle comprises:

accessing the typing context handle through the “this” pointer of the instance.

24. The computer program product of claim 15 wherein the typing context data structure is appended to a virtual table associated with the generic class.

25. An execution engine for executing parametrically polymorphic code and dynamically generating typing context data associated with a typing-context-relevant-code-point being executed within a typing context in a dynamic execution environment, the execution engine comprising:

5 a read module encountering the typing-context-relevant-code-point in the typing context during execution of the program;

a handle module identifying a typing context handle associated with the typing context, the typing context handle referencing a typing context data structure associated with the typing context;

10 a computation module computing the typing context data associated with the typing-context-relevant-code-point;

an allocation module allocating a field in the typing context data structure associated with the typing-context-relevant-code-point; and

15 a recording module recording the typing context data in the field of the typing context data structure.

26. A method of dynamically generating typing context data associated with a typing-context-relevant-code-point being executed within a typing context in a dynamic execution environment, the method comprising:

encountering the typing-context-relevant-code-point in the typing context during
5 execution of the program;
identifying a typing context handle associated with the typing context, the typing context handle referencing a typing context data structure associated with the typing context;
computing the typing context data associated with the typing-context-relevant-code-point;
allocating a field in the typing context data structure associated with the typing-context-relevant-code-point; and
10 recording the typing context data in the field of the typing context data structure.

27. A method of accessing typing context data associated with a typing-context-relevant-code-point being executed within a typing context in a dynamic execution environment, the method comprising:

encountering the typing-context-relevant-code-point in typing context during execution;
5 identifying a typing context handle associated with the typing context, the typing context handle referencing a typing context data structure associated with the typing context; and
retrieving from the typing context data structure the typing context data associated with the typing-context-relevant-code-point.

28. The method of claim 27 wherein the typing-context-relevant-code-point executes a type test on an instance of a generic class, the typing context data includes a resource type descriptor defining the exact type of the instance, and the method further comprises:

performing the type test based on the resource type descriptor associated with the typing-context-relevant-code-point.

29. The method of claim 27 wherein the typing-context-relevant-code-point executes an allocation of an instance of a generic class, the typing context data includes a resource type descriptor defining the exact type of the instance, and the method further comprises:

creating the instance of the generic class based on the resource type descriptor associated with the typing-context-relevant-code-point, wherein the instance is of the exact type.

30. The method of claim 27 wherein the typing-context-relevant-code-point calls a generic method, the typing context data includes another typing context handle, and the method further comprises:

passing the other typing context handle referencing the typing context data to the generic method as a hidden parameter.

31. The method of claim 27 wherein the identifying operation comprises:
retrieving the typing context handle from a stack frame.

32. The method of claim 27 wherein the typing-context-relevant-code-point is executed within an instance of a generic class and the identifying operation comprises:

retrieving a first pointer to the instance; and

retrieving the typing context handle via a second pointer, a second pointer being relative

- 5 to the first point and referencing the typing context handle associated with the instance.

33. The method of claim 27 wherein the computing operation comprises:

retrieving the typing context data associated with the typing-context-relevant-code-point
from a global hash table.

47

34. A computer readable medium having stored thereon a typing context data structure for defining an exact type associated with a typing-context-relevant-code-point within an instance of a generic class during execution in a dynamic execution environment, the data structure comprising:

- 5 a virtual table data portion associated with the generic class;
- a typing context portion associated with the generic class, the typing context portion being accessible through the virtual table portion and having one field storing an exact type associated with the typing-context-relevant-code-point.

35. The computer readable medium of claim 34 wherein the typing context portion of the typing context data structure includes a plurality of fields, each field storing an exact type associated with a typing-context-relevant-code-point encountered during execution in the dynamic execution environment.

36. The computer readable medium of claim 34 wherein the typing context portion of the typing context data structure includes a plurality of fields, each field storing a typing context handle with a typing-context-relevant-code-point encountered during execution in the dynamic execution environment.